# 밈코인 인덱싱용 멀티 프로바이더 아키텍처 (초안)

목표: 토큰 발굴(Discovery) → 체인/네트워크 판별 → 체인별 최적 RPC로 라우팅 → 로깅/리트라이/페일오버까지 포함한 안정적 수집 파이프라인

## 1) 전체 흐름도

[Keyword: "pepe" 등]  
 |  
 v  
[Dune API ➊] -----(대체/보완 소스)-----> [DexScreener API, CoinGecko, Birdeye(솔라나), DefiLlama]  
 | |  
 | v  
 | [Token Candidates: {symbol, addr, chain?}]  
 v  
[Token Resolver]  
 - 체인 판별(힌트: CA 포맷/길이, 체인 전용 익스플로러 조회, 멀티체인 인덱스 API)  
 - 없으면 브루트 방식: 각 체인 RPC에 eth\_getCode/실패시 다음  
 |  
 v  
[Routing Layer]  
 - 체인→프로바이더 매핑  
 ETH: Primary=Alchemy, Secondary=Infura  
 Base/Arbitrum/OP: Primary=Alchemy/QuickNode, Secondary=Infura  
 Solana: Primary=QuickNode, Secondary=Birdeye/공식 RPC  
 BNB/opBNB: Primary=Moralis/QuickNode, Secondary=Public Gateway  
 |  
 v  
[Ingestion Workers]  
 - 표준화 이벤트 스키마(Transfers, Swaps, LP, Metadata)  
 - 배치/스트리밍(WS 가능 체인은 스트림)  
 - 재시도/백오프/아이들 포켓 큐  
 |  
 v  
[Storage]  
 - OLTP: Postgres(원천 로그)  
 - OLAP: DuckDB/BigQuery(지표·백테스트)  
 |  
 v  
[Monitoring]  
 - 체인별 에러율, 지연, 누락 토큰 알람  
 - 프로바이더별 성공률/쿼터/비용 대시보드

## 2) 체인→프로바이더 매핑 (예시)

* **Ethereum (메인넷)**: Alchemy(주), Infura(백업)
* **Base / Arbitrum / Optimism (L2)**: Alchemy 또는 QuickNode(주), Infura(백업)
* **Solana**: QuickNode(주), Birdeye API+공식 RPC(보조)
* **BNB Chain / opBNB**: Moralis 또는 QuickNode(주), Public RPC(보조)

운영 팁: 같은 체인에서도 최소 2개 RPC를 설정하고, 호출마다 라우팅 정책(70/30 로드밸런싱 + 실패 시 페일오버)을 둡니다.

## 3) 라우팅 정책 (권장)

* **우선순위 + 타임아웃**: Primary 2.5s 내 미응답 → Secondary로 즉시 전환
* **회로차단기(Circuit Breaker)**: N회 연속 실패 시 일정 기간 강제 격리
* **레이트리밋 분산**: 70%/30% 가중 라운드로빈
* **헬스체크**: 주기적 eth\_blockNumber/getHealth 프로빙 → SLA 점수화

## 4) Token Resolver 세부

1. **주소 패턴**으로 1차 분기
   * Solana: base58 (44자 등), EVM: 0x40바이트 고정
2. **익스플로러 핑**
   * EVM: Etherscan/Arbiscan/OPscan/ZkSync Explorer 등 API or head 요청
   * Solana: Birdeye/Jupiter/솔스캔 API 조회
3. **RPC 브루트 체크**
   * 후보 체인 리스트에 대해 getCode/getProgramAccounts 등 최소 호출로 배포 여부 확인

## 5) 스키마 표준화 (핵심 이벤트)

* token\_transfers(chain, tx\_hash, block\_time, from, to, amount, token\_addr, decimals, usd\_px, source\_provider)
* dex\_swaps(chain, tx\_hash, block\_time, pool, base, quote, amount\_in, amount\_out, usd\_px, dex\_id)
* pools\_liquidity(chain, pool, token0, token1, tvl\_usd, lp\_token, fee, dex\_id)
* tokens\_meta(chain, address, symbol, name, decimals, project\_links, risk\_flags)

## 6) 운영 체크리스트

* 체인별 Primary/Secondary 키 발급 및 쿼터 알림 연결
* 라우팅 레이어 환경변수(우선순위, 타임아웃, 백오프) 분리
* Dune API 크레딧·rpm 모니터링 연동
* 누락 토큰 자동 재시도 큐
* 비용 대시보드(크레딧/호출/GB-스캔/응답행)

## 7) 단계적 롤아웃

1. **Phase A**: ETH+SOL+BNB 기본 수집(단일 프로바이더)
2. **Phase B**: L2(Base/Arb/OP) 추가 + 라우팅/페일오버 적용
3. **Phase C**: 데이터 품질 지표화(커버리지/지연/누락) + 알람 체계
4. **Phase D**: 비용·성능 최적화(쿼리 캐시, 배치 크기, 압축)

*초안입니다. 체인·프로바이더·요금제는 실제 선택에 맞춰 업데이트 예정.*